



RESEARCH

Open Access



Automatic hand phantom map generation and detection using decomposition support vector machines

Huaiqi Huang^{1,2*} , Claudio Bruschini² , Christian Antfolk³, Christian Enz², Tao Li¹, Jörn Justiz¹ and Volker M. Koch¹

*Correspondence:
huaiqi.huang@epfl.ch
¹ BME Lab, Institute
for Human Centered
Engineering, Bern University
of Applied Sciences,
Quellgasse 21, 2502 Biel,
Switzerland
Full list of author information
is available at the end of the
article

Abstract

Background: There is a need for providing sensory feedback for myoelectric prosthesis users. Providing tactile feedback can improve object manipulation abilities, enhance the perceptual embodiment of myoelectric prostheses and help reduce phantom limb pain. Many amputees have referred sensation from their missing hand on their residual limbs (phantom maps). This skin area can serve as a target for providing amputees with non-invasive tactile sensory feedback. One of the challenges of providing sensory feedback on the phantom map is to define the accurate boundary of each phantom digit because the phantom map distribution varies from person to person.

Methods: In this paper, automatic phantom map detection methods based on four decomposition support vector machine algorithms and three sampling methods are proposed, complemented by fuzzy logic and active learning strategies. The algorithms and methods are tested on two databases: the first one includes 400 generated phantom maps, whereby the phantom map generation algorithm was based on our observation of the phantom maps to ensure smooth phantom digit edges, variety, and representativeness. The second database includes five reported phantom map images and transformations thereof. The accuracy and training/ classification time of each algorithm using a dense stimulation array (with 100×100 actuators) and two coarse stimulation arrays (with 3×5 and 4×6 actuators) are presented and compared.

Results: Both generated and reported phantom map images share the same trends. Majority-pooling sampling effectively increases the training size, albeit introducing some noise, and thus produces the smallest error rates among the three proposed sampling methods. For different decomposition architectures, one-vs-one reduces unclassified regions and in general has higher classification accuracy than the other architectures. By introducing fuzzy logic to bias the penalty parameter, the influence of pooling-induced noise is reduced. Moreover, active learning with different strategies was also tested and shown to improve the accuracy by introducing more representative training samples. Overall, dense arrays employing one-vs-one fuzzy support vector machines with majority-pooling sampling have the smallest average absolute error rate (8.78% for generated phantom maps and 11.5% for reported and transformed phantom map images). The detection accuracy of coarse arrays was found to be significantly lower than for dense array.



Conclusions: The results demonstrate the effectiveness of support vector machines using a dense array in detecting refined phantom map shapes, whereas coarse arrays are unsuitable for this task. We therefore propose a two-step approach, using first a non-wearable dense array to detect an accurate phantom map shape, then to apply a wearable coarse stimulation array customized according to the detection results. The proposed methodology can be used as a tool for helping haptic feedback designers and for tracking the evolution of phantom maps.

Keywords: Phantom map, Support vector machines, Sensory feedback, Hand amputee, Machine learning, Active learning

Background

Although the dexterity of hand prostheses has made significant progress in the past decades, there is still no or limited sensory feedback in commercial prostheses [1, 2]. To our knowledge, the only commercial prosthesis equipped with sensory feedback is the Vincent Evolution 2 hand from the Vincent Systems GmbH [3]. This hand has only one vibrator, providing very limited feedback. A survey conducted by Pylatiuk et al. showed that upper arm amputees would like to have sensory feedback integrated in the hand prosthesis [4]. Providing sensory feedback can not only improve the functionality of the prosthesis, but also enhance the body ownership feeling of the amputees. Another benefit is to relieve phantom limb pain [5].

There are several ways to provide amputees with tactile sensory feedback. The methods can be roughly divided into two main categories: invasive and non-invasive feedback. The invasive approach stimulates the central nervous system using cortical electrodes [6] or the peripheral nervous system using either cuff electrodes [7, 8] or transversal intrafascicular multichannel electrodes [9]. Non-invasive feedback systems apply stimuli on the surface of the skin. The stimuli can be electrical currents (electrotactile) [10–14], vibrations (vibrotactile) [11] or mechanical pushing (mechanotactile) [15, 16] on the skin to elicit sensations.

Many amputees have referred sensation of their lost arm on their remaining stump, called phantom map (examples of phantom maps are shown in Fig 1a). Phantom map could serve as an area to provide sensory feedback. A phantom map is a region on the body that can evoke a sensation of the lost hand. Surveys have shown that 80–90% of amputees develop a phantom map immediately after amputation [17]. While half of those amputees keep stable long-term phantom maps [17], most of the time, the hand phantom maps are present on the face or on the remaining stump. Pressure applied on one area of the phantom map gives the sensation that it comes from a specific finger or an area of the amputated hand.

The dominant theory regarding the phantom map formation is the reorganization of the cortical topography. In the Penfield map, the hand area is bordered by the upper arm and the face. When the hand is amputated, these two regions (upper arm and face) invade the area representing the hand, thereby forming the phantom map [18].

Previous works have demonstrated the feasibility of providing non-invasive sensory feedback on the hand phantom maps [19–21]. One of the benefits of providing sensory feedback on the phantom map is its high spatial resolution. We have indeed observed that the phantom map area has a smaller two-point discrimination distance than the contralateral upper arm, for example. It has also been reported that

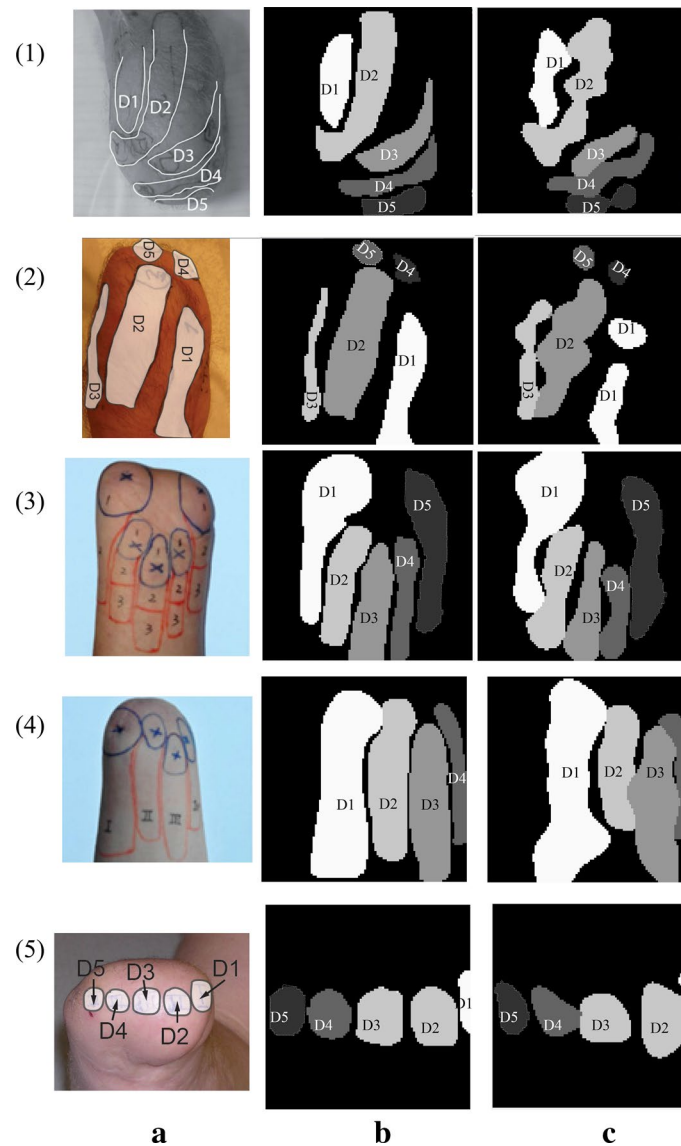


Fig. 1 Reported phantom map image classification examples. Examples of **a** reported phantom map images [13, 23, 27, 28], **b** processed and down-sampled phantom maps, and **c** predicted phantom maps using OVO-SVM and 2×2 majority pooling

somatosensory feedback (feedback applied on the phantom map) has a higher discrimination accuracy and requires lower mental workload when the amputees are required to recognize the stimulation pattern [19].

The phantom map distribution and sensitivity vary from individual to individual because of the “uncontrollable muscle and nerve reorganization” which takes place after the amputation [17, 19]. The phantom map shape can also change over time [18].

Finding the hand phantom map distributions of individual amputees is of great importance in order to provide efficient stimulation patterns and take full advantage of the high spatial resolution provided by the phantom map.

Current phantom map detection methods are still quite rudimentary, using palpation and then drawing the phantom map directly on the skin of the amputee [22, 23]. This detection method is inconsistent and inaccurate. It is difficult to record detailed phantom map distribution and to track the evolvement of each individual phantom map. Moreover, to the best of our knowledge, no phantom map database exists at present.

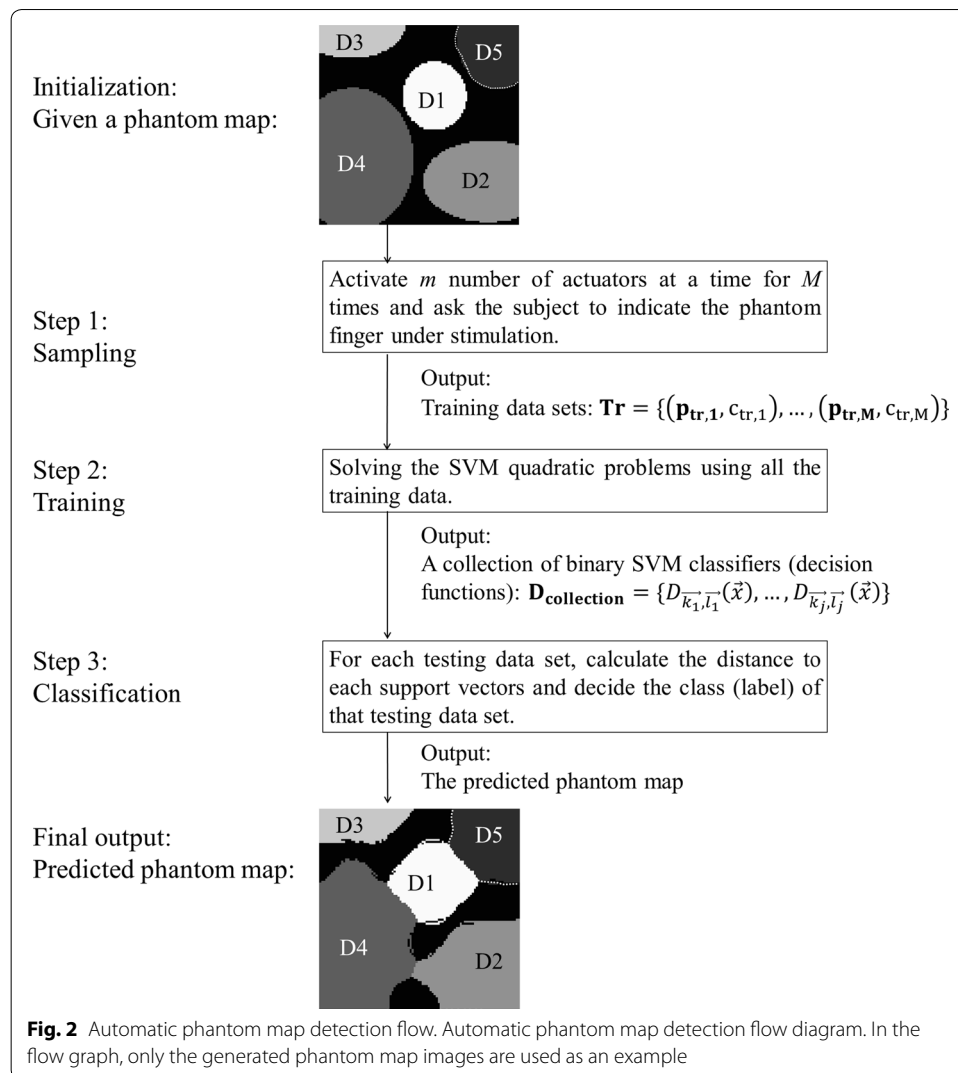
Current sensory feedback applied on phantom maps normally uses one actuator (vibrator or pusher) per phantom finger [20]. In this approach, a rough phantom map distribution detection is sufficient. However, when a dense stimulation array is applied on the phantom map, a more accurate phantom map distribution estimation is needed. The WiseSkin project, for example, aims at providing richer feedback to upper arm amputees by incorporating dense, miniaturized sensory nodes in a hand glove, and using wireless transmission to convey the sensory data to a processing unit [24]. The latter generates signals to activate a corresponding dense actuator array placed on the phantom map of the amputee.

Previous work

An automatic phantom map detection method has been proposed in our previous work [24, 25]. The proposed method collects a small amount of phantom map distribution data of an amputee and uses this data and different algorithms to generate a detailed graphical phantom map. An example of automatic phantom map detection using SVM algorithms is shown in Fig. 2. This approach, which was tested on simulated phantom map models, can help feedback designers to customize the stimulation array and potentially increase the haptic vocabulary. It can also help to find an optimized place for electromyography (EMG) electrodes to avoid interactions between EMG electrodes and sensory feedback arrays. The proposed algorithms included two non-machine learning based (group testing and adaptive edge finding [24]) and two machine learning based algorithms (neural network [24] and support vector machine (SVM) [25]). The simulation results showed that support vector machine based algorithms have higher overall accuracy than the other tested algorithms. They were however based on simplified ellipse-based phantom map models and only their overall error rates were compared, without a detailed training and classification cost analysis (i.e. analysis of the algorithm sensitivity for a series of figures of merit, and training or classification time).

The general approach in this paper is similar to the flow in previous work (Fig. 2), consisting of three stages: sampling, training, and classification. However, this paper substantially extends the previous work in all stages and proposes four multi-class SVM algorithms for automatic phantom map detection, complemented by fuzzy SVM and active learning to further increase the detection accuracy. More realistic and detailed contour phantom map models were generated to test the algorithms using the specifications of fine grained stimulation arrays. Refined metrics were used to evaluate the effectiveness of the algorithms, including on the performance of a “realistic” coarse stimulation array. A time cost and shifting error analysis was included as well.

This paper is organized as follows: the hand phantom map databases, including a novel contour phantom map model generation and reported phantom map processing, are introduced in “[Hand phantom map databases](#)” section, then three data sampling methods are presented in “[Sampling methods](#)” section. The proposed support vector machine



algorithms with and without fuzzy membership functions or active learning are detailed in “[Support vector machine](#)” section. The accuracy and timing aspects of the proposed algorithms are presented “[Results](#)” section and the results are discussed in “[Discussion](#)” section. Conclusions are provided in “[Conclusion](#)” section.

Hand phantom map databases

Two databases are introduced in this section to test the phantom map detection algorithms: the first database consists of generated phantom maps using a contour model (“[Hand phantom map model generation](#)” section), whereas the second database is based on processed and transformed reported phantom map images (“[Database from reported phantom map images](#)” section).

Hand phantom map model generation

In this subsection, we describe the generation process which we employed to produce realistic phantom maps using a contour model (Fig. 2: Initialization). The distribution

and sensitivity of phantom maps do vary individually. From the descriptions of the reported phantom maps and interviews with amputees, it can for example be observed that phantom maps have clear and smooth edges [19, 23]. There can be repeated phantom digit representations (one phantom finger has more than one non-connected area represented on the phantom map) [19]. Some amputees have a complete phantom map, meaning that all five phantom fingers exist, while others only have partial phantom maps (one or more phantom fingers are missing). Furthermore, it is also observed that when several phantom digits are touched simultaneously, the amputee can distinguish all the digits that are being touched.

To simplify the model, it is assumed that there is no overlap between phantom digits. Considering the average size of the remaining stump and the typical minimum two-point discrimination distance, the phantom map was modeled as a 100×100 matrix A . Each element in the matrix is assigned a number from [0, 1, 2, 3, 4, 5], representing no phantom sensation, phantom thumb, phantom index, phantom middle, phantom ring, and phantom little finger, respectively. After having selected the actual phantom map types (number of phantom fingers—either 5 or 10—and completeness/incompleteness, a contour model was used to generate the individual phantom maps. The generation algorithm starts by randomly selecting 4–5 points within an $a \times b$ window (for 5 finger phantom maps, $0 < a, b \leq 60$ and for 10 finger phantom maps, $0 < a, b \leq 45$). The values of a and b were determined empirically to provide reasonable phantom map shapes and a wide range of phantom sensation coverage; for example, when too large it was very difficult for the contour algorithm to converge and/or generate balanced and representative phantom maps. The selected data points are connected by an active snake contour model [26]. All the elements included in the contour edge are assigned the same finger number, starting from 1. Then the generation algorithm continues selecting data until all the needed phantom fingers are assigned (Fig. 3).

The generated phantom maps are then used to test the performance of the proposed detection algorithms.

The size of the phantom sensation area (the area on the remaining stump that can evoke the sensation of the lost fingers) varies from person to person. Thus, we define a variable called ‘phantom sensation coverage (C_{PS})’ to describe the ratio of the phantom sensation area against the remaining stump

$$C_{PS} = \frac{A_{\text{Phantom fingers}}}{A_{\text{Stump area}}}, \quad (1)$$

where $A_{\text{Phantom fingers}}$ is the total phantom finger area, and $A_{\text{Stump area}}$ is the whole stump area, $A_{\text{Stump area}} = 100 \times 100$.

The phantom map model generation method provide the possibility to adjust the phantom sensation coverage range (Fig. 4), select between complete and partial phantom maps, and control the total number of generated phantom digit representations (Fig. 5). Examples of generated phantom map models are shown in Fig. 6.

Database from reported phantom map images

To further validate the proposed algorithms, we also used five reported phantom map images from the literature [13, 23, 27, 28] to build a second phantom map database.

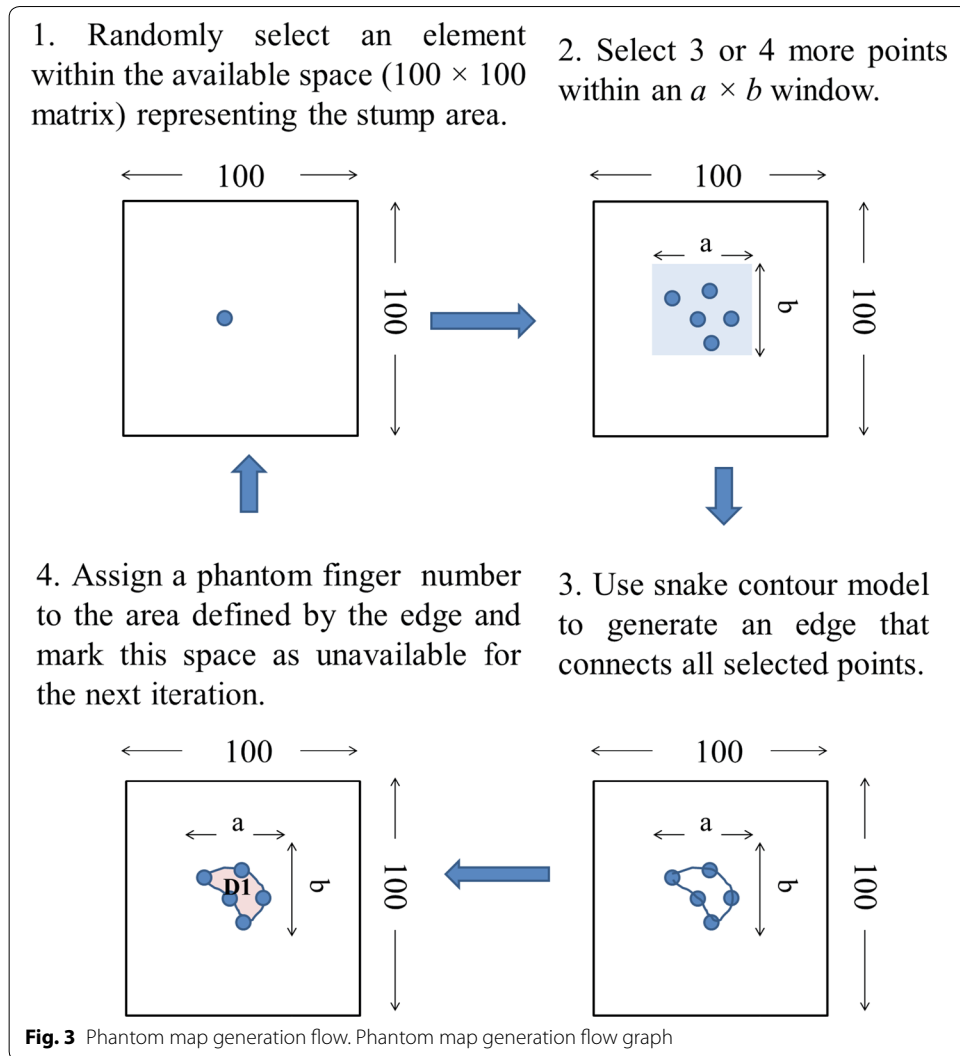


Fig. 3 Phantom map generation flow. Phantom map generation flow graph

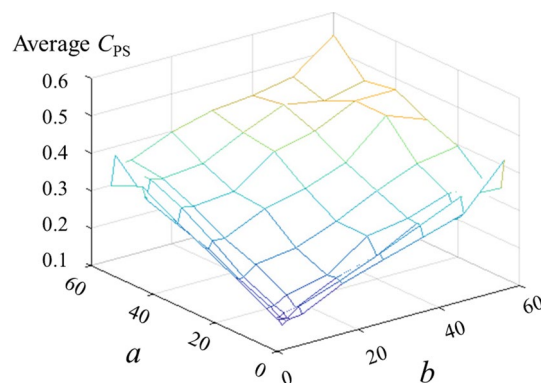
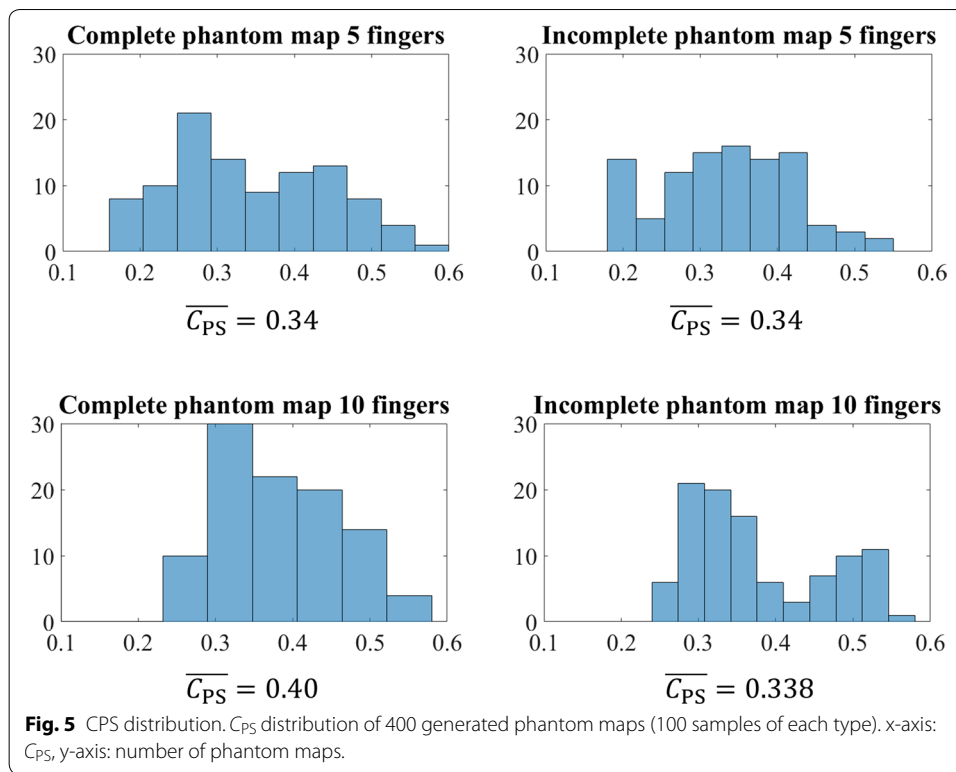


Fig. 4 CPS controlled by a and b . Phantom sensation coverage control: average C_{PS} of 5 finger phantom maps generated by varying a and b within $0 < a, b \leq 60$

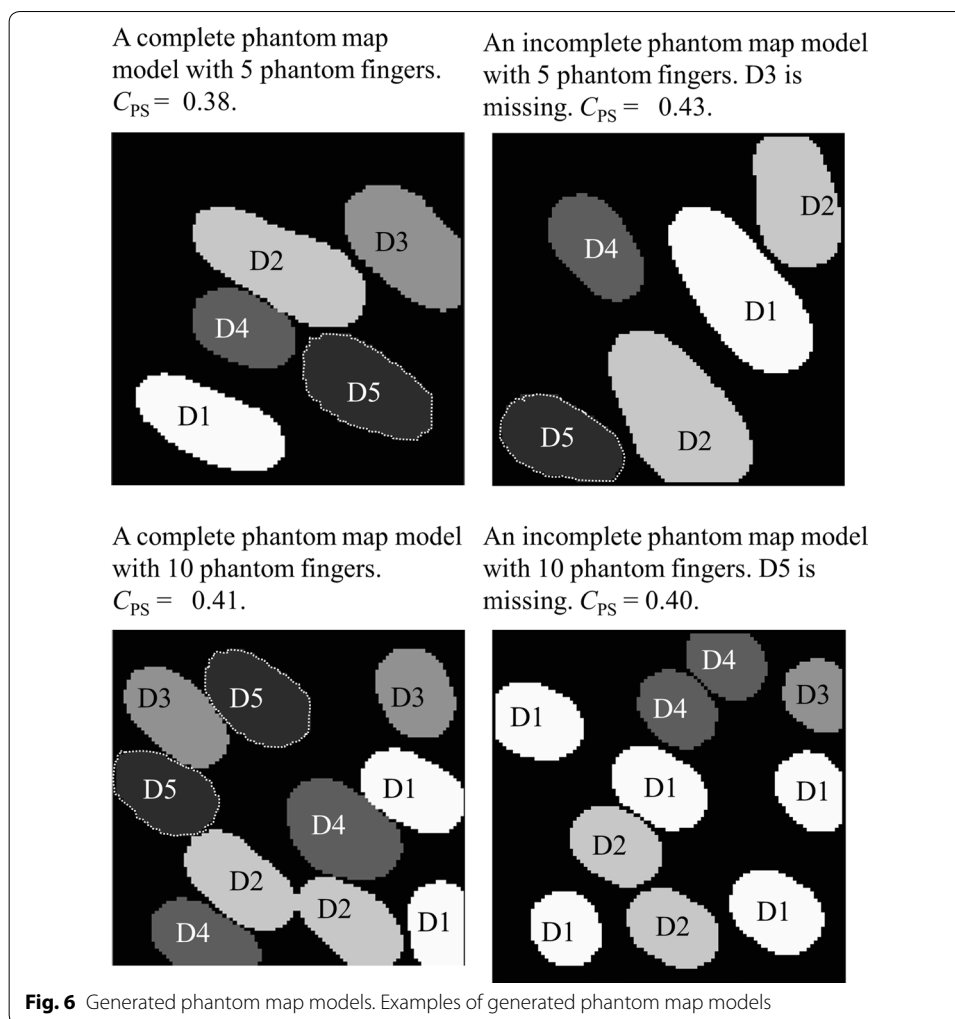


To digitize the reported phantom map images, the edge of each phantom finger in the reported phantom map images (Fig. 1a) was outlined in Illustrator (Adobe Illustrator CC, United States) and each phantom finger area is assigned a color. Then each Illustrator processed image is imported into MATLAB 2017b (The MathWorks, Inc., United States) and down-sampled into a 100×100 matrix (Fig. 1b), with each color mapped to its corresponding grey scale value. The compressed matrix (image) is used for classification. The corresponding predicted phantom maps using OVO-SVM and 2×2 majority pooling are presented in Fig. 1c.

The digitized phantom maps are then transformed into a group of images using rotation, scaling, shearing, translation, and barrel or pin cushion transformation. For rotation, each digitized reported phantom map image is rotated between 0° and 360° for every 5° . For scaling, both proportional scaling and one-dimensional scaling are used. The scaling factor ranges from 10 and 100%. For the translation, both single direction and bi-directional translation are used. The shear factor ranges from 0 to 1. For barrel or pin cushion transformation, the amplitude of the cubic term varies between -0.01 to 0.01 . Examples of the transformed images are shown in Fig. 7.

Sampling methods

In our current work, we apply machine learning algorithms to accurately detect phantom map shapes with limited number of sampled points, referred as samples in the rest of the paper. For machine learning, selecting representative training data is essential, especially in applications where labeling is expensive. From our experience working with amputees, it can be observed that (a) the amputee can give a clear

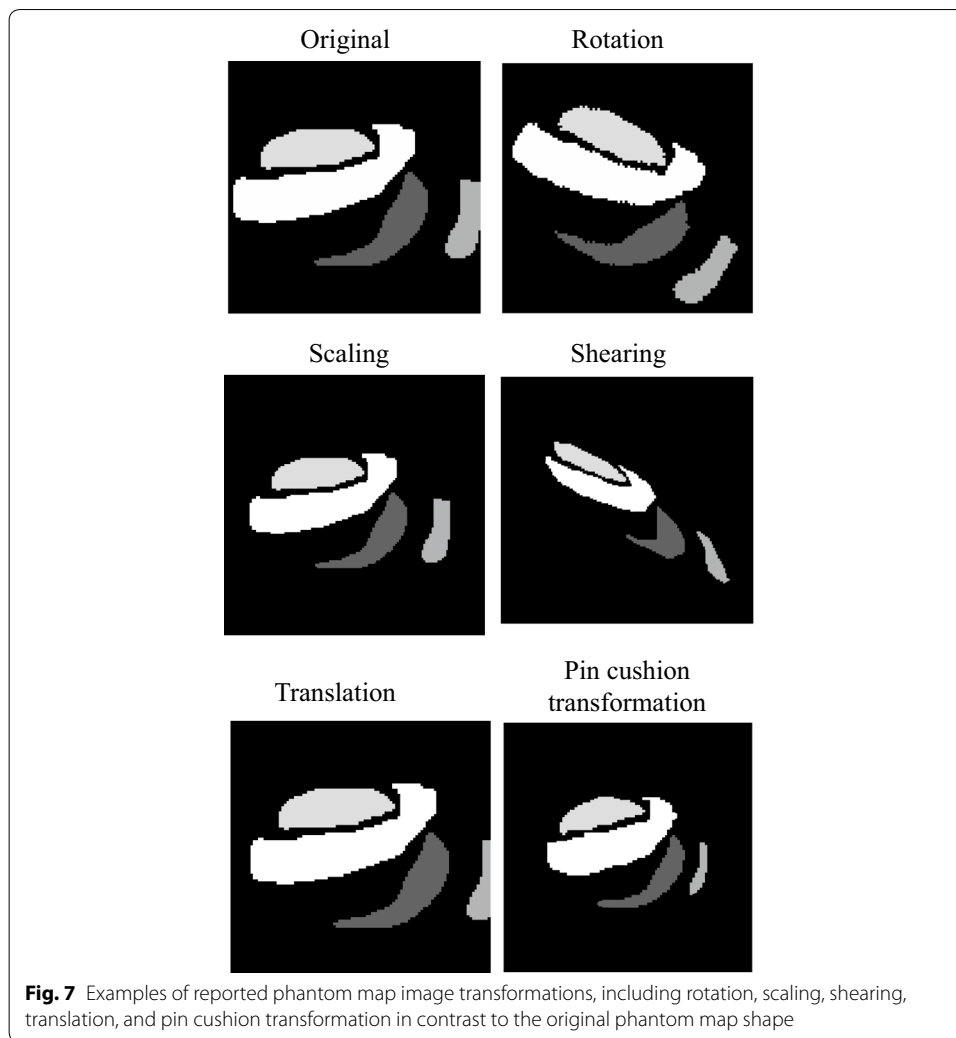


response to the location of the stimuli (i.e., clearly identify which phantom finger has been touched), and (b) when the stimulation falls across several phantom fingers, the amputee can still indicate which finger felt stronger.

Taking these elements into account, three different sampling methods for SVMs are proposed and explained in this section (Fig. 2: Step 1: Sampling): random sampling, systematic sampling, and majority pooling sampling. Although their effectiveness is explored below on simulated data, the sampling protocols are designed in such a way that they are also to be applicable in future clinical tests.

Random sampling (RS)

Random sampling consists of randomly picking m data sets and labeling them individually (Fig. 8a). The m data sets will be used for training the support vector machine algorithms.

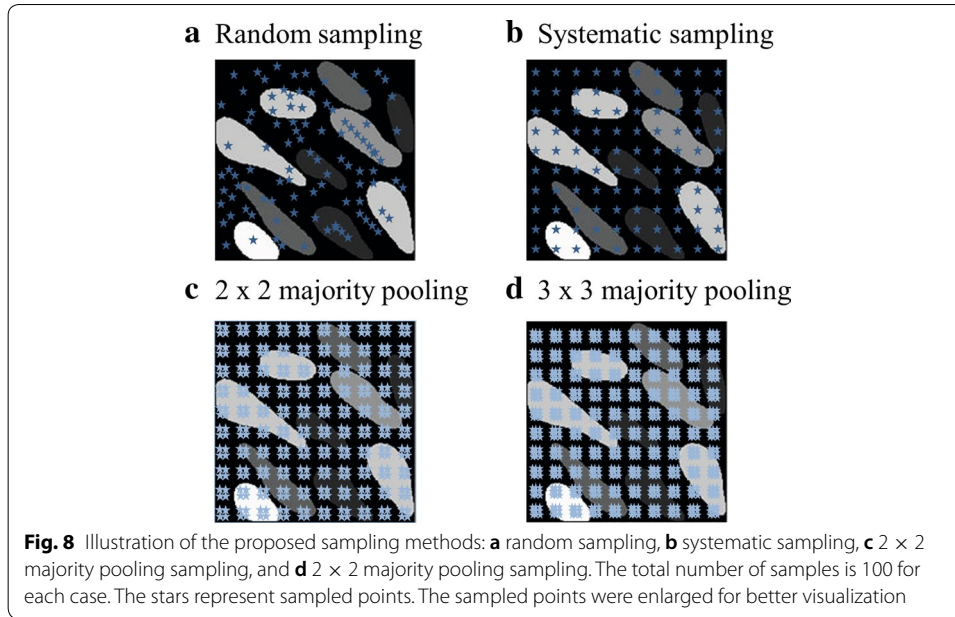


Systematic sampling (SS)

Instead of randomly choosing the query data point, the whole phantom map region is evenly divided into a regular grid. Each grid point is a sampling point (Fig. 8b).

Majority pooling sampling (MPS)

The idea of majority-pooling comes from the max-pooling concept of convolutional neural networks. Applying pooling can result in more compact representations and higher robustness to noise [29]. First, the algorithm defines a set of non-overlapping rectangular windows W_i each containing $p \times q$ sampling points, where $W_i \subset A_{\text{stump}}$ area and $i \in [1, 2, \dots, M]$, M being the number of training data sets. Then the corresponding phantom finger numbers of all the data points within W_i are collected. In clinical practice the subject responds which finger(s) he or she feels being touched, and if more than one phantom finger is within the stimulation region, the amputee has to choose which one feels stronger. In this study, all the points within the selected window will be labeled as the one the amputee chooses, i.e.,



$$\forall i, W_i = \text{Mo}(W_i), \quad (2)$$

where Mo represents mode operation, which selects the value which occurs most frequently in the window.

Examples of majority pooling sampling are shown in Fig. 8c, d. Majority-pooling can introduce errors because all the data points within a pooling window are labeled according to the majority vote. A pooling induced error E_{MP} is therefore defined as

$$E_{\text{MP}} = \frac{N_E}{N}, \quad (3)$$

where N_E is the number of wrongly labeled training data due to pooling and N is the total number of training data sets.

Support vector machine

After selecting the training samples, the support vector machines (SVMs) need to be trained using the acquired samples (Fig. 2: Step 2: training). After the training step, the SVMs can be used to classify the phantom map shape distribution (Fig. 2: Step 3: Classification).

Support vector machine basics

A SVM is a non-probabilistic, parametric binary linear classifier, based on the maximum margin principle. The basic idea behind a SVM is to minimize the classification error rate while maximizing the geometric margin between two classes [30].

In a binary-class classification problem, given M training data sets: $\mathbf{T}_r = \{(\mathbf{p}_{\text{tr},1}, c_1), (\mathbf{p}_{\text{tr},2}, c_2), \dots, (\mathbf{p}_{\text{tr},M}, c_M)\}$, where $\mathbf{p}_{\text{tr},i} \in \mathbf{R}^n$, $i = 1, 2, \dots, M$, $\mathbf{p}_{\text{tr},i}$ is the feature of the i th training data set, \mathbf{R}^n is the feature space, n is the feature dimension, and c_i is the training class label in the i th training data set, whereby $c_i \in \{-1, +1\}$, $i = 1, 2, \dots, M$ (outputs of Fig. 2: Step 1). The SVM training consists in solving the following optimization problem:

$$\begin{aligned}
& \min \quad \frac{1}{2} \|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^M \zeta_i, \\
& \text{s.t.} \quad c_i(\boldsymbol{\omega}^T \mathbf{p}_{\text{tr}} + b) \geq 1 - \zeta_i, \\
& \text{and} \quad \zeta_i \geq 0, \quad i = 1 \dots M,
\end{aligned} \tag{4}$$

where b is the bias of the hyperplane, C is the penalty parameter, and ζ_i is the slack variable.

Because the relationship between class labels (phantom digit number) and attributes (the location of sampling points) is non-linear, a non-linear kernel function is used to map the input space into a feature space for higher classification accuracy. In this paper, a radial basis function (RBF) kernel is chosen [31], because it maps the input space into Hilbert space (an infinite hyperplane) and provides more flexibility [32]. More details about the kernel function and SVM parameter tuning can be found in [Appendix A](#).

Training and classification data definition for automatic hand phantom map detection

As mentioned before, SVMs use collected training data sets, including training features and training classes to find support vectors (a selected number of training data). Then SVMs use the support vectors to assign each testing feature a class. The training data sets collection \mathbf{Tr} is defined as

$$\mathbf{Tr} = \{(\mathbf{p}_{\text{tr},1}, c_{\text{tr},1}), \dots, (\mathbf{p}_{\text{tr},M}, c_{\text{tr},M})\}, \tag{5}$$

where $M = 100$ when using random and systematic sampling, and $M = 100 \times p \times q$ when using majority pooling sampling ($p \times q$ being the pooling size), $\mathbf{p}_{\text{tr},i}$ is the training feature (the position of the data points in a phantom map), with $\mathbf{p}_{\text{tr},i} = (x_i, y_i)$, x_i and y_i being the coordinates of point i in a phantom map matrix M , and c_i is the class label ($c_i \in \{0, 1, 2, 3, 4, 5\}$, $i = \{1, 2, \dots, M\}$).

The collection of testing features is defined as

$$\mathbf{P}_{\text{tst}} = \{\mathbf{p}_{\text{tst},1}, \dots, \mathbf{p}_{\text{tst},10000}\}, \tag{6}$$

whereas the collection of testing classes is defined as

$$\mathbf{C}_{\text{tst}} = \{c_{\text{tst},1}, \dots, c_{\text{tst},10,000}\}, \tag{7}$$

where $c_{\text{tst},1} \in [0, 1, 2, 3, 4, 5]$. For each individual phantom map, M selected points are sampled and used to train the SVMs. Because the distribution of phantom map is different from person to person, each individual phantom map needs its own training and classification.

Multi-class support vector machine

A SVM is intrinsically a binary classifier. Two approaches have been proposed to extend binary SVM classifiers to multi-class SVM classifiers. The first approach, proposed by Crammer and Singer, is a direct method, which treats the multi-class classification problem as a large ‘constrained optimization problem with a quadratic objective function’ [33]. The second approach is to decompose a multi-class classification problem into a collection of binary classification problems. The direct approach is slow and involves a complex optimization problem. The decomposition approaches generally offer good performance and are easier to implement [34].

There are mainly four decomposition methods: one-vs-all (OVA), one-vs-one (OVO), the directed acyclic graph (DAG), and the binary tree (BT) method.

One-vs-all support vector machine (OVA-SVM)

The principle of an OVA-SVM is to train each class against all the rest of the classes. When training the j th class, the j th class is assigned positive labels, while the others are assigned negative labels. After training all the binary classifiers, the final class is determined by the highest output value.

One-vs-one support vector machine (OVO-SVM)

For an OVO-SVM, all the combinations of class pairs need to be trained. For a k -class classification problem, each binary classifier determines a preferred class. After training all the $k(k-1)/2$ classifiers, the class that has the most votes wins. Although an OVO-SVM needs to train more binary SVMs, the training data contained in each subset is smaller. Thus, the training time needed for each individual SVM is smaller, compared to that of an OVA-SVM.

Directed acyclic graph support vector machine (DAG-SVM)

The training of a DAG-SVM is the same as OVO-SVM. In the classification phase, it follows a rooted binary directed acyclic graph (DAG). For a k -class classification problem, this graph has $k(k-1)/2$ internal nodes and k leaves. Each node is a binary classifier D_{ij} . Each leaf represents a class. k is the total number of classes [35]. In our application, the DAG would be [0 1 2 3 4 5] for a complete phantom map, where 0 represents no phantom sensation and 1–5 correspond to the five phantom fingers. Then each binary SVM chooses the preferred class between the first and the last class of the list. The non-preferred class is then deleted from the list. This procedure continues until a final class decision is reached.

Binary tree support vector machine (BT-SVM)

A BT-SVM is constructed based on a binary tree structure. Each internal node is a binary SVM. During the training phase, half of the remaining training data are assigned positive labels, the other half negative labels. The main goal of a BT-SVM is to reduce the number of binary classifiers, thus decreasing the needed training and classification time.

Table 1 lists the number of binary SVMs required for a k -class classification problem.

Table 1 Multi-class SVM classifiers: number of binary SVMs required for the four main decomposition methods

Method	Number of binary SVMs required for a k -class classification problem	Number of SVMs required to classify a complete phantom map
OVA	k	6
OVO	$\frac{k(k-1)}{2}$	15
DAG	$\frac{k(k-1)}{2}$	15
BT	$2\log_2 k$	5

Fuzzy support vector machines

Although the classification accuracy of a SVM is very high, it is highly influenced by noise and outliers [36]. Fuzzy SVM (FSVM) was proposed to increase the robustness of a conventional SVM by applying a fuzzy membership function to the training data sets [37]. The fuzzy membership function is used to reformulate the SVM so that noisy input points contribute less in training the decision surface. In our classification application and within the proposed model, the noise mainly comes from majority pooling sampling (“Majority pooling sampling (MPS)” section). In order to reduce the effect of pooling-induced errors, a step fuzzy membership function f_c is therefore proposed:

$$f_c = \begin{cases} 1 & \text{when } \forall i \text{ and } j, i \neq j, S_i = S_j, \\ \alpha & \text{otherwise,} \end{cases} \quad (8)$$

where i and j are the element indices in a pooling window, S_i and S_j are the phantom sensation labels of the i th and j th element, with $S_i, S_j \in [0, 1, 2, 3, 4, 5]$, and α is a constant.

In this approach, the penalty parameter C in (4) becomes an array:

$$\mathbf{C} = C_{\text{const}} \times f_c, \quad (9)$$

where C_{const} is the penalty parameter value in a conventional SVM.

Active learning support vector machine

Based on the consideration that the human phantom map detection process is likely gradual and adaptive, we decided to apply active learning as well to the four decomposition SVMs. Active learning is able to query the candidate data interactively, using a specified rule (called query strategy) and sequentially adding new data for labeling and contribution to the training [38]. One of the most widely used query strategies is uncertainty sampling [39, 40], whereby margin sampling and its variations, especially multi-class level uncertainty, have shown good performance when combined with SVMs [39]. Moreover, in order to achieve faster training, batch-mode active learning is often used, whereby a group of instances is added at a time. Furthermore, diversity criteria are often employed for query data selection to ensure the representativeness of selected instances. In the current study, we applied angle-based diversity SVMs [41, 42].

Different initial and batch sizes with and without diversity criteria were employed in this study. During preliminary testing and parameter tuning, we found out that (a) the diversity criteria did not improve the classification accuracy, and (b) the initial size 80 with batch size 2 achieved the highest classification accuracy. In the following sections, only results using the aforementioned testing configuration will be presented.

Results

In the absence of a wearable dense stimulation array, the aforementioned algorithms were tested on the two databases detailed in “Hand phantom map model generation” section, containing 400 generated phantom map models (Fig. 5), as well as processed and transformed reported phantom map images, respectively. Both the phantom map models and the phantom map detection methods were implemented in MATLAB 2017b (The MathWorks, Inc., United States). The program was running on a HP laptop with an Intel core i5-4300 CPU@1.90GHz.

Table 2 Simulation setup for the different sampling methods

	Sampling methods	
	RS and SS	MPS (pooling size = $p \times q$)
# Training data sets per phantom map model	100	$100 \times p \times q$
# Testing data sets per phantom map model	10,000	
Number of phantom maps in each database		
Phantom map database	Type	Number of phantom map images
# Generated phantom maps	Complete 5	100
	Complete 10	100
	Incomplete 5	100
	Incomplete 10	100
	Reported (original)	5
# Transformed reported phantom map images	Rotation	72×5
	Scaling	10×5
	Shearing	20×5
	Translation	10×5
	Barrel or pin cushion	20×5

We have opted for a total of 100 samples for the sampling phase based on initial testing results, which highlighted that a sampling size smaller than 100 led to a large increase in error rate, while the accuracy increase was not significant above this size. We also believe this number to be realistic for a dense array. Indeed, from our experience of testing with amputees, the average time needed to stimulate and get the response is 15–30 s. In practice, when the total number of samples is 100, 25–30 min are needed to complete data collection (the active time involving an amputee). This time is deemed acceptable without causing fatigue nor adaptation.

The overall simulation setup is summarized in Table 2. The accuracy (“Accuracy” section) and timing (“Timing” section) results are presented and discussed below.

Accuracy

The accuracy are presented by six types of metrics, defined in “Accuracy metrics definition” section. The influences of two types of stimulation arrays: dense and coarse, SVM parameters, sampling methods, different SVM algorithms, and socket shifting on accuracy are presented.

Accuracy metrics definition

To evaluate the accuracy of the phantom map detection algorithms, six types of metrics are defined: absolute error rate (E_A), functional error rate (E_F), redundancy error rate (E_R), insufficiency error rate (E_I), precision error rate (E_P), and phantom sensation coverage ratio (R_{PSC}) between a generated phantom map and its corresponding predicted phantom map.

The general error rate E is defined as

$$E = \frac{\sum_{i=1}^N f_i(c_{i,a}, c_{i,p})}{N}, \quad (10)$$

$$\text{where } f_i(c_{i,a}, c_{i,p}) = \begin{cases} 1 & \text{when } c_{i,a} \neq c_{i,p}, \\ 0 & \text{when } c_{i,a} = c_{i,p}, \end{cases}$$

$c_{i,a}$ is the real label of the i th testing set, $c_{i,p}$ is the predicted label of the i th testing set, and N is the number of testing data sets for E_A , E_F , E_R , and E_I , and the number of testing data sets containing phantom sensation for E_P .

- For E_A , $c_{i,a} \in \{0, 1, 2, 3, 4, 5\}$, $c_{i,p} \in \{0, 1, 2, 3, 4, 5\}$. E_A measures the fraction of all misclassified data points of a predicted phantom map.
- For E_F , $c_{i,a} \in \{1, 2, 3, 4, 5\}$, $c_{i,p} \in \{1, 2, 3, 4, 5\}$. E_F measures the fraction of points belonging to one phantom finger which are falsely classified into another finger, leading to a functional error (wrong finger stimulation when providing sensory feedback). ($E_F = 1 - \text{recall}$ [43]).
- For E_R , $c_{i,a} = 0$, $c_{i,p} \in \{1, 2, 3, 4, 5\}$. E_R measures the fraction of points belonging to class 0 (i.e. no phantom sensation) which are wrongly classified into other classes. When providing sensory feedback, these points do not cause mistakes between fingers, but their stimulation is redundant and costs energy without providing useful feedback.
- For E_I , $c_{i,a} \in \{1, 2, 3, 4, 5\}$, $c_{i,p} = 0$. E_I measures the loss of stimulation points which takes place when data points belonging to class 1 to 5 (phantom thumb to phantom little finger) are misclassified as class 0 (no phantom sensation) and therefore not stimulated.
- For E_P , $c_{i,a} \in \{1, 2, 3, 4, 5\}$, $c_{i,p} \in \{1, 2, 3, 4, 5\}$. E_P is an extension of the *precision* measurement of binary classification [43]; it indicates the fraction of incorrectly classified phantom sensation points with respect to all the phantom sensation points in the generated phantom map.

The relationship of the first four types of error rates is:

$$E_A = E_F + E_R + E_I. \quad (11)$$

E_P is related to E_F :

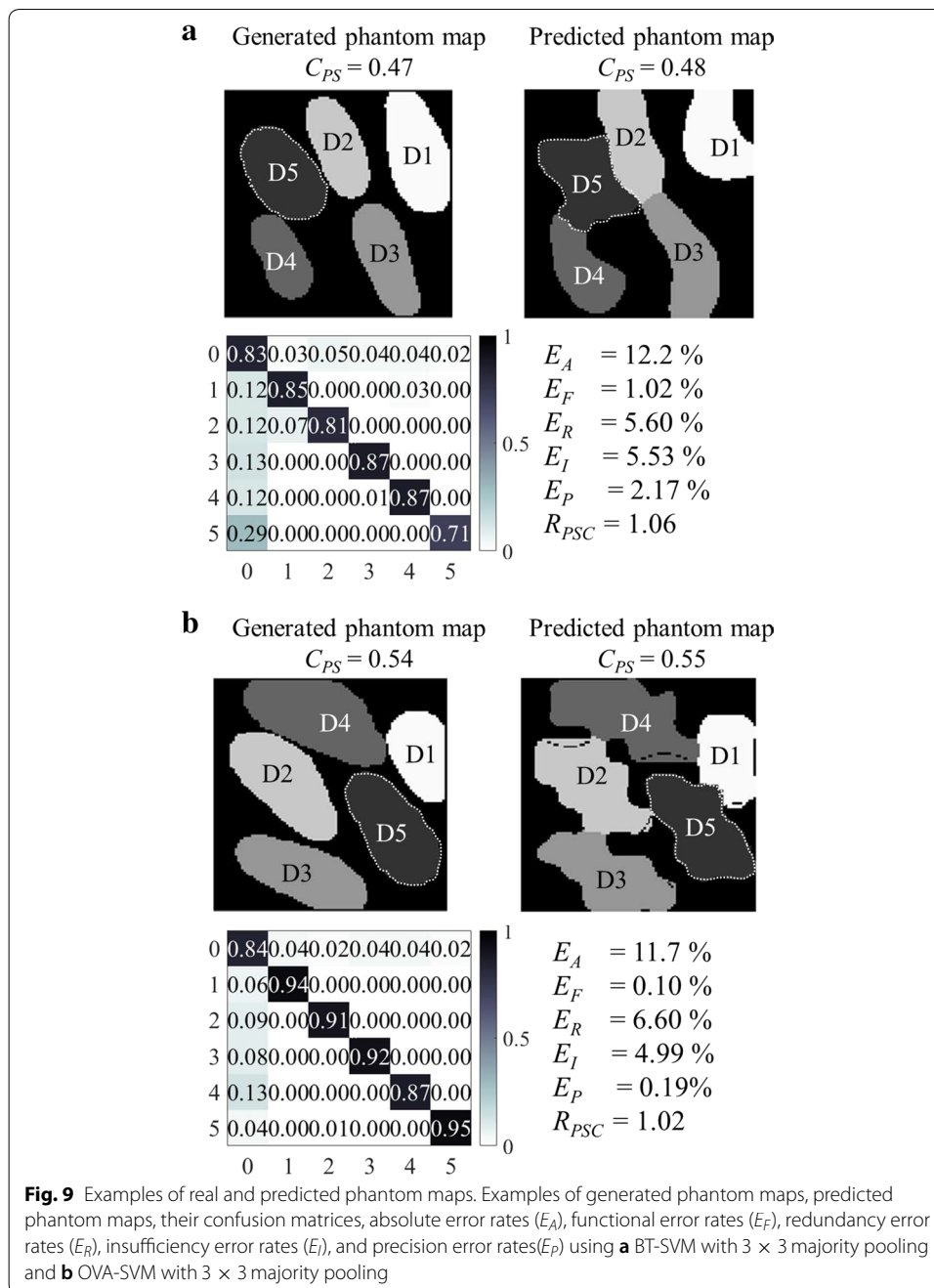
$$E_P = \frac{E_F}{C_{PS}}. \quad (12)$$

R_{PSC} is defined as

$$R_{PSC} = \frac{C'_{PS}}{C_{PS}}, \quad (13)$$

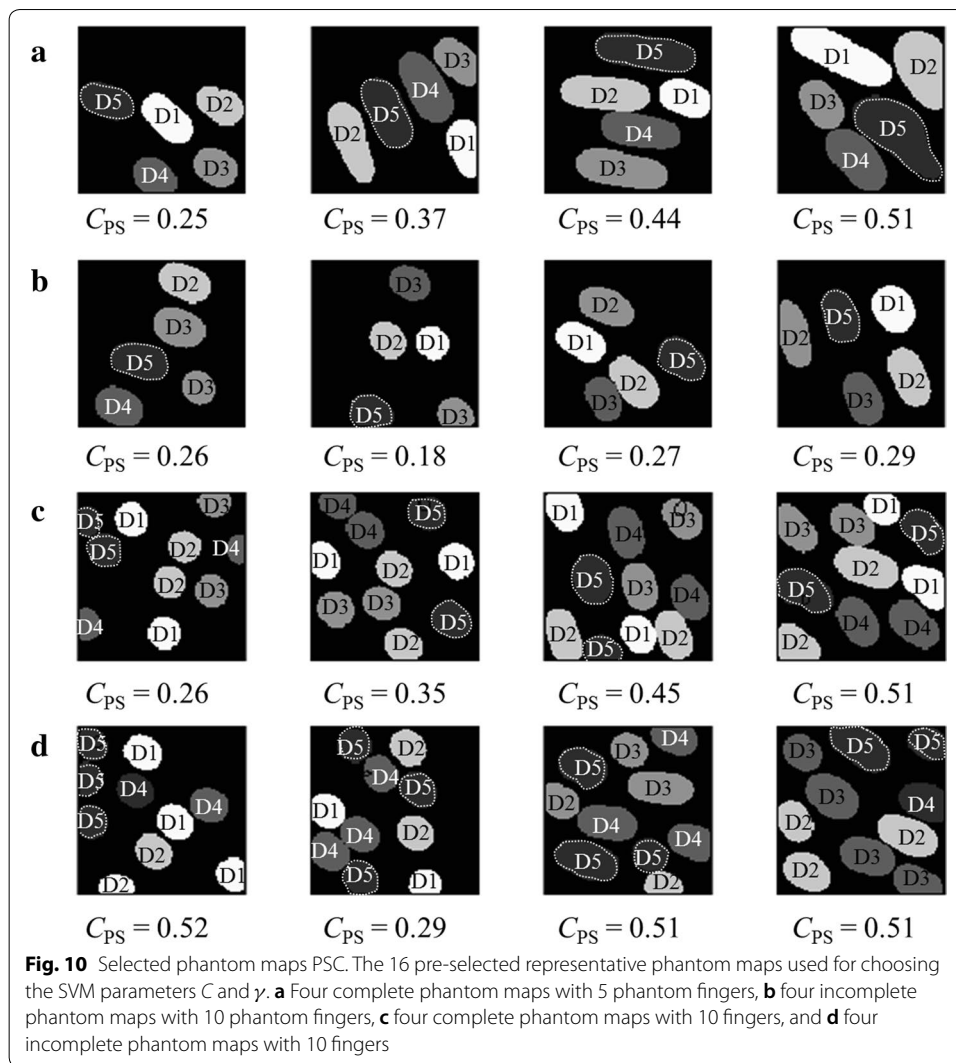
where C'_{PS} is the phantom sensation coverage of the predicted phantom map and C_{PS} is the phantom sensation coverage of the original generated phantom map. R_{PSC} defines the proportion of the predicted phantom map C'_{PS} over the corresponding generated phantom map model (the original C_{PS}) (13).

To demonstrate the defined metrics, Fig. 9 shows examples of generated phantom maps, predicted phantom maps, their confusion matrices and accuracy metrics.



SVM parameters C and γ

For SVM-based methods, the parameter selection exerts a great influence on the classification performance. The involved parameters are the penalty parameter C and the RBF kernel parameter γ . In order to determine the optimal C and γ values, 16 pre-selected representative phantom maps were used (shown in Fig. 10), carrying out a grid search to cover a range of C and γ values ($C \in [10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10, 20, 30, 40, 50]$ and



$\gamma \in [10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10, 20, 30, 40, 50]$). The C and γ pairs that produce the smallest absolute error rate E_A are selected for further evaluation.

Grand average accuracy

The grand average accuracy is the average error rate over all the phantom maps in one database. The grand average accuracy and phantom sensation coverage ratio of generated phantom maps and reported, as well as their transformed phantom map images are presented separately in Fig. 11.

The accuracy results obtained by using the reported phantom map images showed similar trends as those obtained by using the generated phantom maps. For example, both types of phantom maps benefited from majority-pooling sampling, OVO decomposition

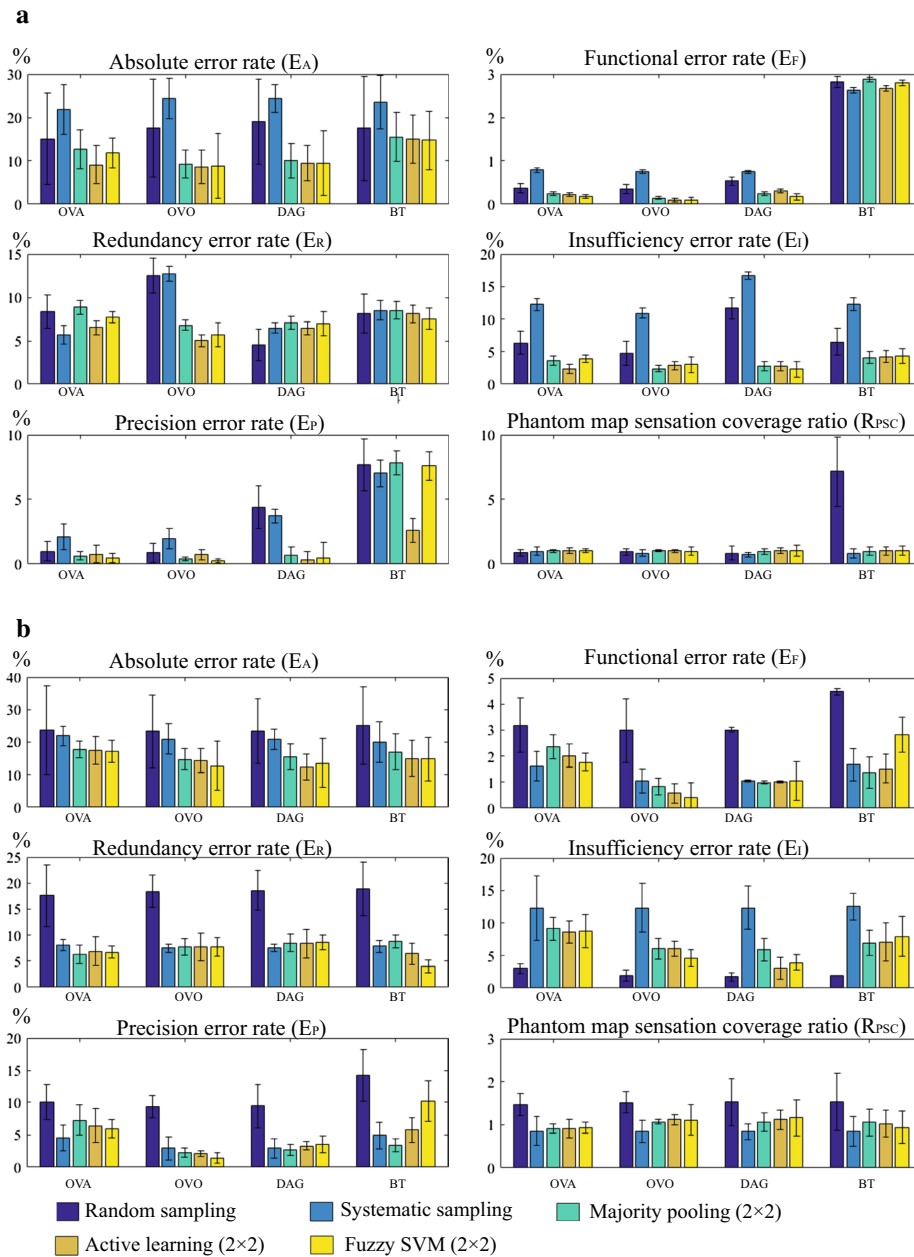


Fig. 11 Grand average accuracy using dense array. Grand average error rates and phantom sensation coverage ratios over **a** all 400 generated phantom maps and **b** five reported phantom map images and their corresponding transformed images. For 2×2 majority pooling, $\bar{E}_{MP} = 5.35\%$ for generated phantom maps and $\bar{E}_{MP} = 4.27\%$ for reported phantom map images. The grand average accuracy is influenced both by the sampling methods and SVM algorithms used. For both generated and reported phantom maps, OVO-SVM produces the smallest error rate. Even though the absolute error rate (E_A) for reported phantom maps is higher than for the generated ones, the more critical metric (function error rate E_F) is still within an acceptable range

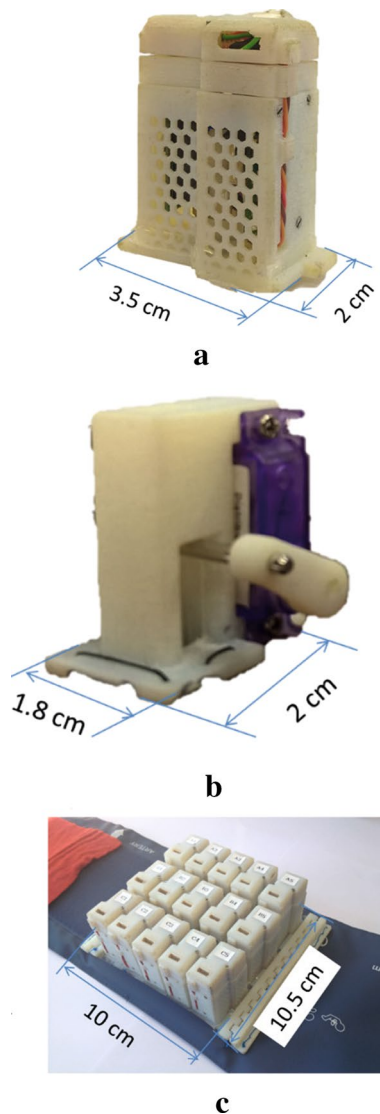


Fig. 12 Stimulation devices. Two types of stimulation devices and an experimental coarse stimulation array. These are primarily used for providing sensory feedback for upper limb amputees. **a** Hybrid (vibrotactile and mechanotactile) stimulation device, **b** mechanotactile stimulation device

architecture, and added fuzzy logic or active learning to the penalty parameters. The following analyses (“[Discussion](#)” section) are therefore applicable to both types, although we will focus mainly on the discussion of the generated data. Quantitatively speaking, the error rates of the reported phantom maps are slightly higher than those of the generated ones, for all the algorithms and sampling methods tested. This could be explained by the lower average phantom sensation coverage of the former.

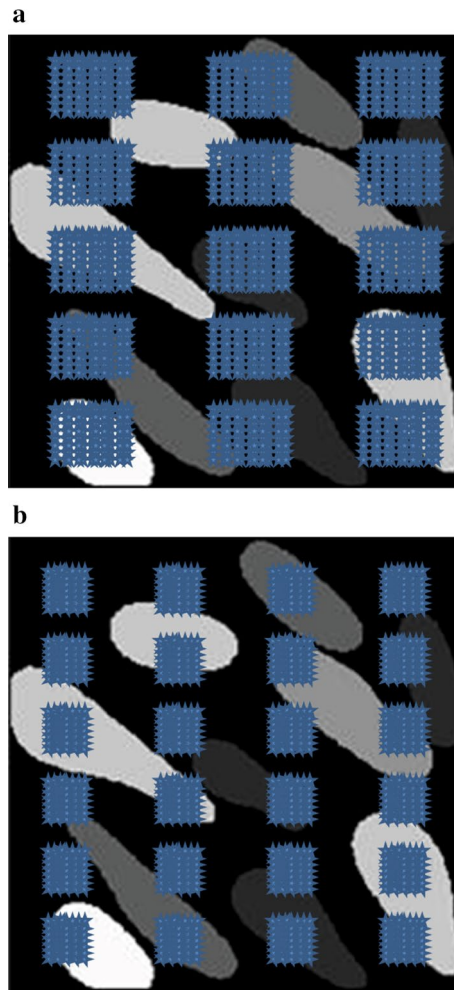


Fig. 13 Coarse array sampling methods. Graphical representations of coarse array sampling. The blue stars represent sampling points. **a** Hybrid stimulation array: 3×5 sampling size with 15×9 pooling size. **b** Mechanotactile stimulation array: 4×6 sampling size with 7×7 pooling size

Phantom map detection accuracy using coarse arrays

The potential performance using coarse stimulation arrays designed in our lab is also evaluated. These stimulation arrays are designed primarily to provide sensory feedback for upper limb amputees. Figure 12a is a hybrid stimulation actuator combining a servo motor and an eccentric rotating mass vibrator. The minimal contact size is fixed by the vibrator (153 mm^2). Fig. 12b is a servo motor based mechanotactile vibrator; the arm and pin are 3D printed and the contact size is controllable. A 3×5 hybrid actuator array (Fig. 12c) [44] or a 4×6 mechanotactile actuator array can fit on the remaining stump of an amputee and are currently being investigated.

The average phantom map area is roughly 100 cm^2 . The minimum actuator contact sizes for mechanotactile and hybrid are 100 and 153 mm^2 , respectively. Given that in a simulation scenario the pooling size reflects the physical contact size, the corresponding minimum pooling sizes $p \times q$ (defined in 4.3) are 15×9 and 7×7 for hybrid and mechanotactile actuators, respectively (Fig. 13).

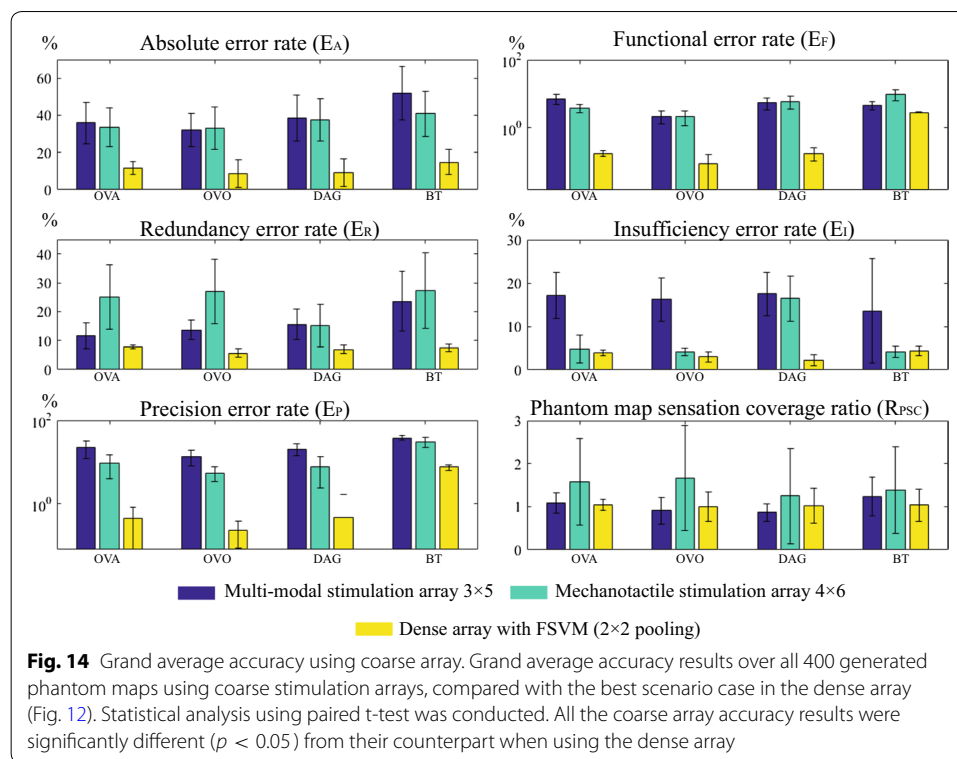


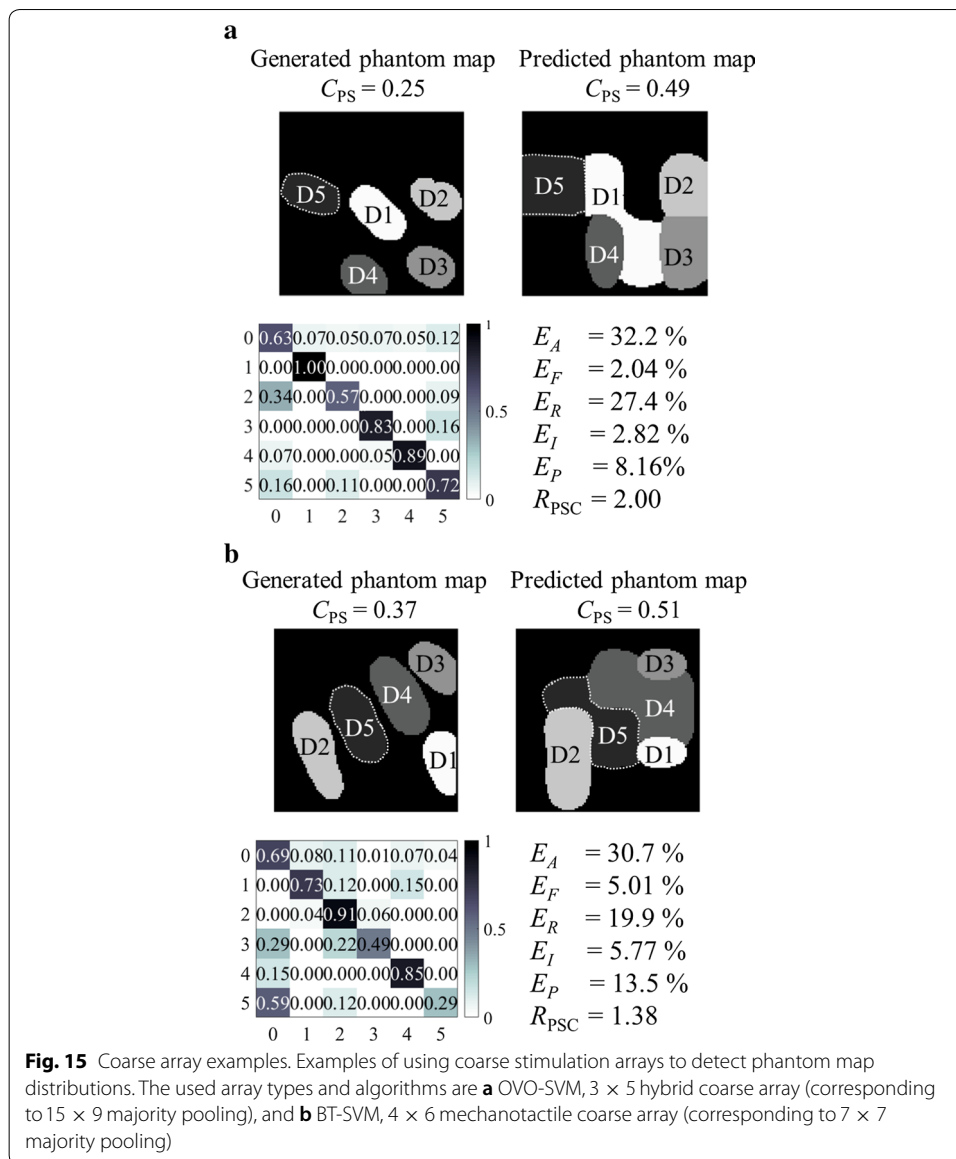
Figure 14 shows the grand average accuracy results of five types of error rates and phantom sensation coverage ratios when using coarse array to detect phantom map shapes, compared with the best case scenario from generated phantom maps and reported and transformed phantom maps. Fuzzy SVMs were also evaluated; the overall performance was however not significantly improved when using FSVMs ($p = 0.34$ when using the paired t-test to compare the results from standard SVMs and fuzzy SVMs), and the corresponding results are therefore not reported. Fig. 15 shows examples of generated phantom maps, their corresponding predicted phantom maps, the confusion matrices, and six metrics.

Systematic error caused by socket shifting

After detecting the phantom map shape, the stimulation devices will be used to provide sensory feedback. The stimulation devices themselves are embedded in a socket, which the amputees will need to wear on and off on a daily basis. The socket position can therefore shift, for example laterally as shown in Fig. 16. We have therefore simulated the effect of such a movement on the different error rates (E_A , E_F , E_R , E_I , and E_P) in case of the OVO-SVM algorithm applied to a scenario using complete phantom maps with 5 fingers each, for different expected shift levels (Fig. 17).

Timing

Different sampling and training methods result in different training (T_t) and classification times (T_c). Table 3 shows the grand average training and classification time using



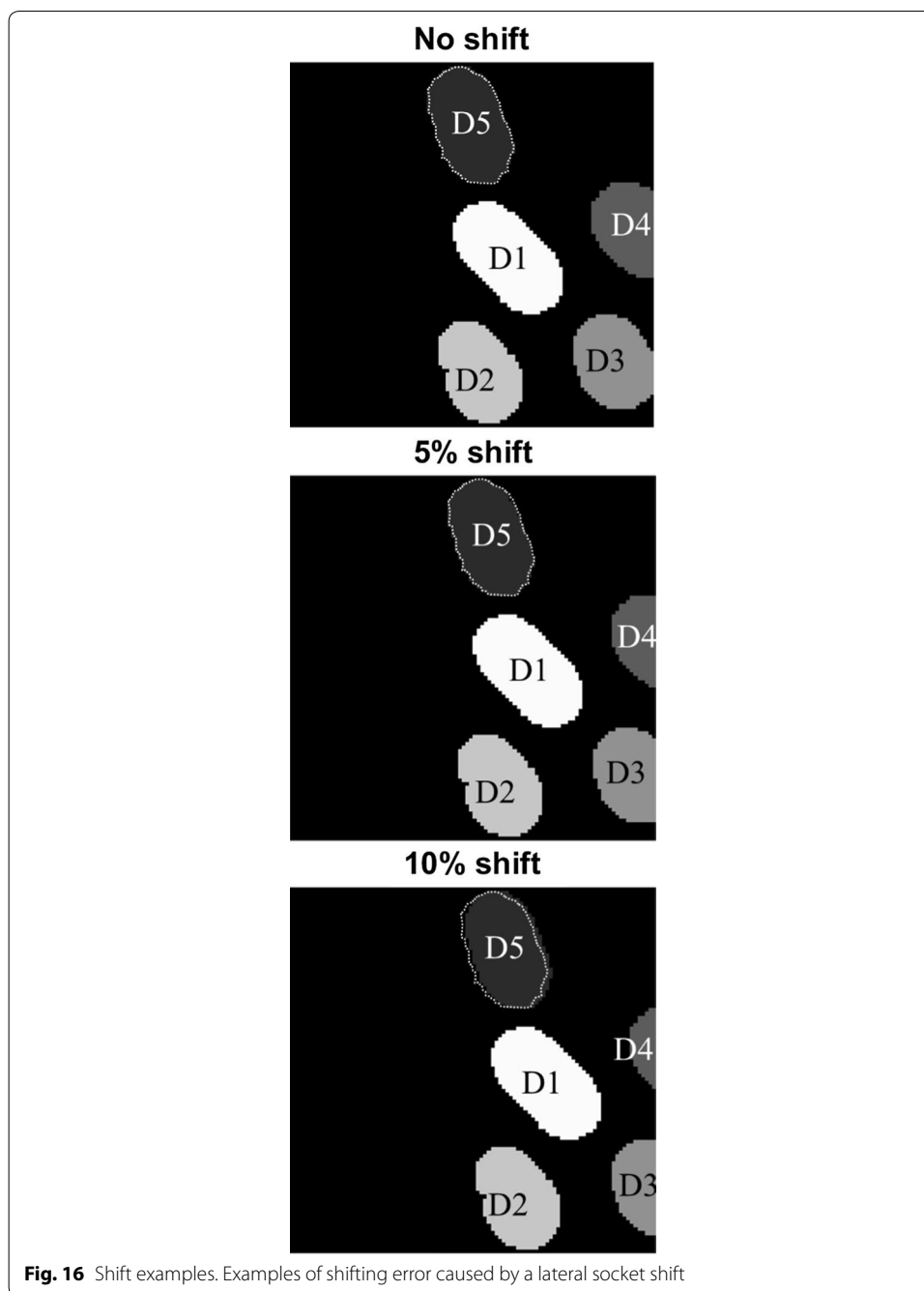
different sampling methods, averaged over all 400 generated phantom maps and calculated for the target ideal dense array as well as for the two examples of coarse stimulation arrays currently under investigation to provide sensory feedback (see also “[Phantom map detection accuracy using coarse arrays](#)” section).

Discussion

The influences of different SVM algorithms and sampling methods on classification accuracies and training and classification time are discussed in this section.

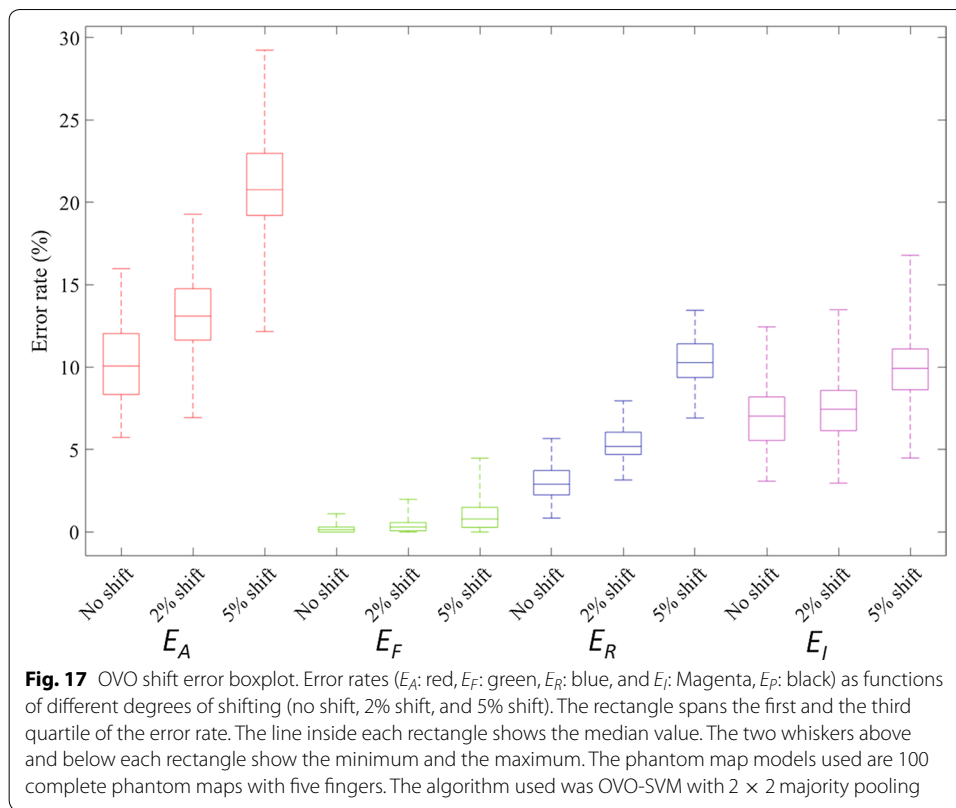
Performance of different sampling methods

In general, random and systematic sampling produce higher absolute error rate E_A (Fig. 11). The reason is that the two sampling methods can not sample enough



representative data points, as illustrated qualitatively in Fig. 8. Unlike majority pooling, which covers a large range of samples, random and systematic sampling only sample 1% of the total data, thus resulting in poor prediction performance. The two methods also produce a high insufficiency error rate E_I (Fig. 11). This means that the two sampling methods cannot fully grasp the trend of a phantom map model.

All four decomposition multi-class SVM algorithms benefit to various degrees from majority pooling sampling (Fig. 11), which results in more training data sets without increasing the active time involving an amputee. For all algorithms, majority pooling



sampling reduces absolute error rate (E_A) and functional error rate (E_F). It was also observed that for the chosen dense array settings (Table 2), 2×2 majority pooling produces the smallest error rates for all five error rate metrics.

However when using majority pooling in other settings, there is a trade-off between the pooling-induced error rate and sampling range. A larger pooling size can produce a larger sampling range coverage, but it introduces more pooling-induced errors or noise. For a particular setting, an optimal pooling size exists.

The influence of different decomposition SVM methods

Overall, OVO-SVM has the smallest absolute error rate (Fig. 11) among the four tested algorithms and the predicted phantom map shapes best represent the original generated phantom map shapes (Fig. 9). The OVO architecture reduces the unclassified regions (compared to the OVA architecture), is less sensitive to unbalanced data sets (compared to the BT architecture), and provides a more thorough evaluation (compared to the DAG architecture).

The error rates of OVA-SVM when using majority pooling are higher than those of OVO-SVM, and lower than those of BT-SVM. The major issue with OVA-SVM is the presence of unclassified regions, as can be seen in Fig. 9b (dashed black lines running within a phantom map finger).

Among the four proposed algorithms, BT-SVM is intrinsically different from the other three multi-class SVMs. All the other three methods classify, to some degree,

one class at a time, whereas a BT-SVM tries to separate a group of classes from another group of classes. When classifying a complete phantom map, the BT-SVM first classifies between class group (0, 1, 2) and class group (3, 4, 5). When then classifying between classes (3, 4, 5), it does not consider the classes (0, 1, 2), but only looks for the largest margin between the classes (3, 4, 5) themselves. This can result in two of the class regions being connected in the predicted phantom map, such as in Fig. 9a.

When using BT-SVM, different tree structures can produce different prediction results, especially when dealing with unbalanced data sets. BT-SVM should theoretically have a faster training and classification speed, however, the fast speed is at a price of degraded performance [45].

The influence of fuzzy logic and active learning

Fuzzy logic and active learning are applied to each decomposition SVM algorithm. A FSVM assigns a fuzzy membership function (8) to each training data set, so that each training data set makes a different contribution in the training process. A FSVM can reduce the influence of pooling induced errors $\overline{E_{MP}}$ (3). Using FSVMs generally increases the detection accuracy when using the 100×100 dense arrays. It was also observed that FSVMs reduce the unclassified region for OVA- and OVO-SVM, as was also reported in previous literature [46, 47]. Active learning also helps to increase the detection accuracy by selecting more representative training data.

Detection accuracy using coarse arrays

Comparing Figs. 11 and 14, it can be observed that the accuracy decreases significantly when using coarse stimulation arrays.

The discussion about the influence of decomposition methods and sampling methods when using a dense array stands true for the coarse array. However, FSVMs do not decrease the error rate when coarse arrays are used. The reason could be that the pooling sizes of coarse arrays are much larger but the sampling density is much smaller, so that the pooling induced error E_{MP} is larger than that of a dense array.

In order to accurately detect a phantom map distribution, a dense array is therefore needed. However, to the best of our knowledge, no wearable dense array (100×100) is readily available. We therefore propose to divide the design of a sensory feedback system into two parts: the first part makes use of a non-wearable dense array to detect the accurate boundaries of a phantom map, then according to the shape distribution, a customized stimulation array with 20–30 actuators can be integrated into a wearable socket.

Systematic error caused by socket shifting

As could be expected, all the error rates increase with the shifting degree (Fig. 17). The absolute error rate E_A increases dramatically when the shifting degree increases (Fig. 17). However, the corresponding increase in the average functional error rate E_F is important in relative terms (from 0.12 to 0.97%), still small in absolute terms. The E_A increase at 5% lateral shifting, for example, is indeed mainly caused by an increase of the redundancy error rate (reaching 10.1%) and of the insufficiency error rate E_I (reaching 9.94%).

In other words, despite the fact that $\overline{E}_A = 21.0\%$ at 5% lateral shift, the more critical error rate \overline{E}_F amounts to less than 1%, which is small enough to be negligible. We can conclude that the function of a stimulation array would be minimally affected by a slight socket misalignment.

Timing

The training and classification times increase substantially with the pooling size, but still stays within an acceptable range. Given the same number of training data sets, the training and classification time were influenced by decomposition architectures. OVO and DAG share the same training process. Under the same conditions, the training times of OVO and DAG are the same. As mentioned in “[One-vs-one support vector machine \(OVO-SVM\)](#)” section, given the same number of training data sets, OVO and DAG-SVM do not require significantly more training time than the other two methods when using random and systematic sampling, and sometimes even less training time when using majority pooling sampling. The classification processes of OVO and DAG-SVM are different. DAG-SVM requires much less classification time than OVO-SVM at the price of a slightly higher absolute error rate (E_A). For active learning, due to the iterative nature, the training time was several times longer than that of standard SVMs and FSVMs, although still less than 1 s (Table 3).

Table 3 Grand average training time \overline{T}_t and classification time \overline{T}_c of all 400 generated phantom maps using a dense array (100 samples) and two coarse (stimulation) arrays (3×5 and 4×6 actuators, corresponding to simulation pooling sizes of 15×9 and 7×7)

Method	\overline{T}_t (ms)	\overline{T}_c (s)	\overline{T}_t (ms)	\overline{T}_c (s)
Dense array				
	OVA		OVO	
RS	35.0	15.9	54.9	33.7
SS	28.6	15.3	47.9	32.7
MP (2×2)	84.2	17.5	79.3	39.6
AL	300	16.1	278	38
	DAG		BT	
RS	54.9	15.8	24.8	5.57
SS	47.9	15.5	19.8	5.30
MP (2×2)	79.3	16.5	48.1	6.05
AL	278	17	239	5.83
Hybrid coarse (stimulation) array, 3×5 actuators				
	OVA		OVO	
MP 15×9	952	34.7	356	46.7
	DAG		BT	
	356	35.8	301.9	9.13
Mechanotactile coarse (stimulation) array, 4×6 actuators				
	OVA		OVO	
MP 7×7	348.3	25.2	196	45.0
	DAG		BT	
	196	43.5	153	9.79

RS random sampling, SS systematic sampling, MP majority pooling sampling, AL active learning, OVA one-vs-all, OVO one-vs-one, DAG direct acyclic graph, BT binary tree

Conclusion

In this paper, three sampling methods and four decomposition multi-class SVMs were proposed for use in automatic phantom map detection.

In the absence of wearable dense stimulation arrays, the accuracy and timing aspects were tested on realistic and flexible generated phantom maps as well as five reported phantom map images and transformations thereof. The phantom map generation algorithm considered different types of phantom maps and introduced parameters to provide a variety of reasonable and representative shapes. The trends of the classification results obtained by the two types of phantom maps are similar. Therefore, the analyses and discussion were applicable for both generated and reported phantom maps.

For the dense stimulation array, we have found that fuzzy OVO-SVM with 2×2 pooling size has the highest classification accuracy and near real-time training speed (less than 1 s training time for both generated and reported phantom maps).

The potential performance using coarse stimulation arrays, designed primarily to provide sensory feedback, was also evaluated and found to be much lower than that of a dense array. Thus, they are unsuitable for refined phantom map shape detection. We therefore propose a two-step approach, using first a non-wearable dense array to detect an accurate phantom map shape, then to apply a wearable coarse stimulation array customized according to the detection results.

To the best of our knowledge, this is the first attempt to apply machine learning algorithms to identify the distribution of phantom maps. The proposed methodology can be used as a tool for helping haptic feedback designers and for tracking the evolvement of phantom maps.

Abbreviations

SVM: support vector machine; RS: random sampling; SS: systematic sampling; MPS: majority pooling sampling; RBF: radial basis function; OVA: one-vs-all; OVO: one-vs-one; DAG: directed acyclic graph; BT: binary tree; FSVM: fuzzy support vector machine.

Authors' contributions

HH engineered the methods and algorithms, analyzed the data and drafted the manuscript. CB contributed to the data analysis and evaluation parameters and co-drafted the manuscript. VK and JJ proposed the idea, and contributed to model building and algorithm development. CA, CE, TL, VK, JJ supervised the study and proofread the manuscript. All authors read and approved the final manuscript.

Author details

¹ BME Lab, Institute for Human Centered Engineering, Bern University of Applied Sciences, Quellgasse 21, 2502 Biel, Switzerland. ² Integrated Circuits Laboratory (ICLAB), École Polytechnique Fédérale de Lausanne (EPFL), Rue de la Maladière 71b, 2002 Neuchâtel, Switzerland. ³ Biomedical Engineering, Lund University, Ole Römers väg 3, 22100 Lund, Sweden.

Acknowledgements

This work is part of the WiseSkin project, which is supported by Nano-Tera and Swiss National Science Foundation (SNSF) under grant No. 20NA21 143070, the Swedish Research Council under grant No. 637-2013-444, and the Promobilia Foundation.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The generation and detection algorithms as well as generated 400 phantom map models, the five reported phantom map images, and the transformed image files will be available upon request.

Consent for publication

Not applicable.

Ethics approval and consent to participate

This study did not involve any subjects. It is not necessary to have ethics approval.

Funding

We declare that the sources of funding, Nano-Tera, Swiss National Science Foundation, and the Swedish Research Council were not involved in the study and the writing of the manuscript.

Appendix A: Radial basis function and SVM parameter tuning

Kernel functions are essential to expand the classification capacity of a linear classifier to handle non-linear data. In this paper, a radial basis function (RBF) kernel

$$K(p_i, p_j) = \exp\left(-\frac{\|p_i - p_j\|^2}{2\gamma^2}\right) \quad (14)$$

is chosen, because it maps the input space into Hilbert space (an infinite hyperplane) and provides more flexibility [32]. In (14), p_i and p_j are the features from the i^{th} and j^{th} training or classification data sets.

In general, the penalty parameter C defines the ‘softness’ of a SVM. A large C gives a high penalty for non-separable points. It tends to overfit, exaggerating minor fluctuations or noise in the data. On the other hand, if C is too small, the SVM tends to underfit, meaning that the SVM is not able to find the main trend of the underlying data. The RBF kernel parameter γ defines the influence of a single training data set: a large γ value implies that a single example exerts great influence on the whole SVM model. Both overfitted SVMs and underfitted SVMs produce poor prediction performance.

Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 30 January 2018 Accepted: 18 May 2018

Published online: 11 June 2018

References

1. Otto Bock. <http://www.ottobock.com/>. Accessed 5 Oct 2016.
2. Touch Bionics: I-Limb. <http://www.touchbionics.com/products/active-prostheses/i-limb-ultra>. Accessed 5 Oct 2016.
3. Vincent systems. <http://vincentsystems.de/en/>. Accessed 5 Oct 2016.
4. Pylatiuk C, Schulz S, Kargov A, Bretthauer G. Two multiarticulated hydraulic hand prostheses. *Artif Organs*. 2004;28(11):980–6.
5. Dietrich C, Walter-Walsh K, Preißler S, Hofmann GO, Witte OW, Miltner WHR, Weiss T. Sensory feedback prosthesis reduces phantom limb pain: proof of a principle. *Neurosci Lett*. 2012;507:97–100. <https://doi.org/10.1016/j.neulet.2011.10.068>.
6. Tabot GA, Kim SS, Winberry JE, Bensmaia SJ. Restoring tactile and proprioceptive sensation through a brain interface. *Neurobiol Dis*. 2015;83:191–8.
7. Ortiz-Catalan M, Håkansson B, Brånemark R. An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs. *Sci Transl Med*. 2014;6(257):257–66.
8. Tan DW, Schiefer MA, Keith MW, Anderson JR, Tyler J, Tyler DJ. A neural interface provides long-term stable natural touch perception. *Sci Transl Med*. 2014;6(257):257–338.
9. Raspopovic S, Capogrosso M, Petrini FM, Bonizzato M, Rigosa J, Di Pino G, Carpaneto J, Controzzi M, Boretius T, Fernandez E. Restoring natural sensory feedback in real-time bidirectional hand prostheses. *Sci Transl Med*. 2014;6(222):222–319.
10. Anani A, Ikeda K, Körner L. Human ability to discriminate various parameters in afferent electrical nerve stimulation with particular reference to prostheses sensory feedback. *Med Biol Eng Comput*. 1977;15(4):363–73.
11. Kaczmarek KA, Webster JG, Bach-y-Rita P, Tompkins WJ. Electrotactile and vibrotactile displays for sensory substitution systems. *IEEE Trans Biomed Eng*. 1991;38(1):1–16.
12. Szeto AY, Saunders FA. Electrocutaneous stimulation for sensory communication in rehabilitation engineering. *IEEE Trans Biomed Eng*. 1982;4:300–8.
13. Chai G, Sui X, Li S, He L, Lan N. Characterization of evoked tactile sensation in forearm amputees with transcutaneous electrical nerve stimulation. *J Neural Eng*. 2015;12(6):066002.

14. Anani A, Körner L. Discrimination of phantom hand sensations elicited by afferent electrical nerve stimulation in below-elbow amputees. *Med Progress Through Technol.* 1979;6(3):131–5.
15. Sensinger JW, Schultz AE, Kuiken TA. Examination of force discrimination in human upper limb amputees with reinnervated limb sensation following peripheral nerve transfer. *IEEE Trans Neural Syst Rehabil Eng.* 2009;17(5):438–44.
16. Kim K, Colgate JE. Haptic feedback enhances grip force control of semg-controlled prosthetic hands in targeted reinnervation amputees. *IEEE Trans Neural Syst Rehabil Eng.* 2012;20(6):798–805.
17. Pirowska A, Wloch T, Nowobilski R, Plaszcwski M, Hocini A, Ménager D. Phantom phenomena and body scheme after limb amputation: a literature review. *Neurologia i Neurochirurgia Pol.* 2014;48(1):52–9. <https://doi.org/10.1016/j.pjnns.2013.03.002>.
18. Ramachandran VS, Hirstein W. The perception of phantom limbs. the DO Hebb lecture. *Brain.* 1998;121(9):1603–30.
19. Zhang D, Xu H, Shull PB, Liu J, Zhu X. Somatotopic feedback versus non-somatotopic feedback for phantom digit sensation on amputees using electrotactile stimulation. *J Neuro Eng Rehabil.* 2015;12(1):1.
20. Antfolk C, D'Alonzo M, Controzzi M, Lundborg G, Rosen B, Sebelius F, Cipriani C. Artificial redirection of sensation from prosthetic fingers to the phantom hand map on transradial amputees: vibrotactile versus mechanotactile sensory feedback. *IEEE Trans Neural Syst Rehabil Eng.* 2013;21(1):112–20.
21. Antfolk C, D'Alonzo M, Rosén B, Lundborg G, Sebelius F, Cipriani C. Sensory feedback in upper limb prosthetics. *Expert Rev Med Devices.* 2013;10(1):45–54.
22. Ehrsson HH, Rosén B, Stocksélius A, Ragnö C, Köhler P, Lundborg G. Upper limb amputees can be induced to experience a rubber hand as their own. *Brain.* 2008;131(12):3443–52.
23. Antfolk C, Björkman A, Frank S-O, Sebelius F, Lundborg G, Rosen B. Sensory feedback from a prosthetic hand based on air-mediated pressure from the hand to the forearm skin. *J Rehabil Med.* 2012;44(8):702–7.
24. Huang H, Enz C, Grambone M, Justiz J, Li T, Unsal O, Koch VM. Data fusion for a hand prosthesis tactile feedback system. In: 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS); 2014. p. 604–7. <https://doi.org/10.1109/BioCAS.2014.6981798>.
25. Huang H, Li T, Antfolk C, Bruschini C, Enz C, Justiz J, Koch VM. Automatic hand phantom map detection methods. In: 2015 IEEE biomedical circuits and systems conference (BioCAS); 2015. p. 1–4.
26. Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. *Int J Comput Vision.* 1988;1(4):321–31.
27. Schmalzl L, Thomke E, Ragnö C, Nilseryd M, Stocksélius A, Ehrsson HH. "Pulling telescoped phantoms out of the stump": manipulating the perceived position of phantom limbs using a full-body illusion. *Front Hum Neurosci.* 2011;5:121.
28. Björkman A, Wijk U, Antfolk C, Björkman-Burtscher I, Rosén B. Sensory qualities of the phantom hand map in the residual forearm of amputees. *J Rehabil Med.* 2016;48(4):365–70.
29. Boureau Y-L, Ponce J, LeCun Y. A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th international conference on machine learning (ICML-10); 2010. p. 111–8.
30. Bartlett P, Shawe-Taylor J. Generalization performance of support vector machines and other pattern classifiers. In: Advances in Kernel methods—support vector learning; 1999. p. 43–54.
31. Scholkopf B, Sung K-K, Burges CJ, Girosi F, Niyogi P, Poggio T, Vapnik V. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Trans Signal Process.* 1997;45(11):2758–65.
32. Amari S-I, Wu S. Improving support vector machine classifiers by modifying Kernel functions. *Neural Netw.* 1999;12(6):783–9.
33. Crammer K, Singer Y. On the algorithmic implementation of multiclass Kernel-based vector machines. *J Mach Learn Res.* 2002;2:265–92.
34. Sabzekear M, GhasemiGol M, Naghibzadeh M, Sadoghi Yazdi H. Improved DAG SVM: a new method for multi-class SVM classification. *Int Conf Artif Intell ICAI.* 2009;2009:548–53.
35. Platt JC, Cristianini N, Shawe-Taylor J. Large margin DAGs for multiclass classification. *Nips.* 1999;12:547–53.
36. Wu Z, Zhang H, Liu J. A fuzzy support vector machine algorithm for classification based on a novel PIM fuzzy clustering method. *Neurocomputing.* 2014;125:119–24.
37. Lin C-F, Wang S-D. Fuzzy support vector machines. *IEEE Trans Neural Netw.* 2002;13(2):464–71.
38. Calma, A., Reitmaier T, Sick B, Lukowicz P, Embrechts M. A new vision of collaborative active learning. 2015. arXiv preprint [arXiv:1504.00284](https://arxiv.org/abs/1504.00284).
39. Tuia D, Volpi M, Copa L, Kanevski M, Munoz-Mari J. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE J Select Topics Signal Process.* 2011;5(3):606–17.
40. Settles B. Active learning literature survey. Madison: University of Wisconsin; 2010. p. 11.
41. Brinker K. Incorporating diversity in active learning with support vector machines. *ICML.* 2003;3:59–66.
42. Demir B, Persello C, Bruzzone L. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Trans Geosci Remote Sens.* 2011;49(3):1014–31.
43. Ting KM. Precision and recall. In: Encyclopedia of machine learning. Berlin: Springer; 2011. p. 781–1.
44. Li T, Huang H, Justiz J, Koch VM. A miniature multimodal actuator for effective tactile feedback: design and characterization. *Procedia Eng.* 2016;168:1547–50.
45. Cheong S, Oh SH, Lee S-Y. Support vector machines with binary tree architecture for multi-class classification. *Neural Inform Process Lett Rev.* 2004;2(3):47–51.
46. Abe S, Inoue T. Fuzzy support vector machines for multiclass problems. In: ESANN. 2002. p. 113–8.
47. Inoue T, Abe S. Fuzzy support vector machines for pattern classification. In: IEEE international joint conference on neural networks IJCNN'01, vol. 2. 2001. p. 1449–54.